

2.41 $-\frac{1}{4}$: 符号位为负, 指数位为-2
答案为 101111010000000000000000

3.44 001 001 001 001 001 001 001 不准确

3.47. 8位 16bit 乘法器
~~加~~来 4个 16位值的和并进行约简
移位和位操作
128 64 32 位加载并自取其最高有效位

for $i=0$ to 15 do

load register $B[\text{bits } 127:0] = \text{sig-in}[i*87:127]$

for $j=0$ to 7 do

eight-lane multiply $C[\text{bits } 127:0] = A * F$

set $D[\text{bits } 15:0] = \text{sum of the four 16-bit values}$

set $D[\text{bits } 31:16] = \text{sum of the}$ in $C[\text{bits } 63:0]$

four 16-bit values in $C[\text{bits } 127:64]$

store $D[\text{bits } 31:0]$ to sig-out E

set $A = A$ shifted 16 bits to the left

set $E = B$ shifted 112 shifts to the right

set $A = A$ or E

set $B = B$ shifted 16 bits to the left

endfor

end for

4.2.1

这个指令使用了指令存储器,既有读写端口,用ALU将Rd和Rs加在一起,数据内存,还有寄存器的写端口

4.2.2

不需要,该指令可以用现有的单元实现

4.2.3

不需要,该指令可以在不添加新控件的情况下实现信号,只需改变控制逻辑

4.4.1

由于I-Mem时间比Add长,所以时钟周期为I-Mem延时:200ps

4.4.2

这条指令关键路径是通过指令寄存器Sign-Extend和Shift-Left-2得到偏移,添加单位计算,新建PC并用Mux选择。

$$200ps + 15ps + 10ps + 70ps + 20ps = 315ps$$

4.4.3

条件分支延时路径与计算分支地址和无条件分支一样。此外,它们有一个很长延时:通过寄存器,Mux,ALU计算PCSrc条件。

关键路径是两者中较长,以及通过PCSrc的路径

4.4.4 PC相关分支

4.4.5 c部分不在,并且只需用于PC-relative分支。注意MIPS实际的无条件分支(bne 0, 0, Label)起到相关作用)所以MIPS语言,没有明确类型指令让该单元位于关键路径上

4.4.6 bne 和 add 中, bne 的关键路径较长, 决定了时钟周期 (add 每条路径 \leq bne 对应的路径) 因此我们看单位如何影响 bne 的关键路径。该单位不在关键路径上, 所以必须增加到了 PC Src 通过寄存器, Mux 和 ALU 的路径。

ALU 是 200ps, Sign-Extend, Shift-Left-2 和 Add 是 95ps

所以 Shift-Left-2 必须增加至少 105ps 才会影响时钟周期

4.7-1

符号扩展: 0000 0000 0000 0000 0000 0000 0000 0000

左移两位: 0001 0001 0001 0001 0001 0001 0001 0001

4.7.2 ALU ~~结果~~: 00 指令: 010100

4.7.3 新 PC: PC+4

4.7.4 WtReg Mux: 2 或 0 ALU Mux: 20 Mem/ALU Mux: X

Branch Mux: PC+4 Jump Mux: PC+4

4.7.5 ALU: -3 和 20 PC+4: PC 和 4 分支: PC+4 和 20x4

4.7.6

4.8.1 流水线: 350ps 非流水线: 1250ps

4.8.2 流水线 1750ps 非流水线: 1250ps

4.8.3 ID: 300ps

4.8.4 35%

4.8.5 65%

4.8.6 多周期是流水线的 $0.2 \times 5 + 0.8 \times 4 = 4.2$ 倍

单周期是流水线的 $1250/350 = 3.57$ 倍

流水线 350ps